def make\_phrase(day):

"""

Sings the song based on the selected day.

Parameters

----------

DataFrame containing columns: 'Day' which is a number

Returns

-------

Text describing the gift for that day.

"""

#Add the column

item\_amounts = [

'a',

'two',

'three',

'four',

'five',

'six',

'seven',

'eight',

'nine',

'ten',

'eleven',

'twelve']

xmas['item\_amount'] = item\_amounts

while day >= 1:

# Select the day

row = xmas.loc[xmas["Day"] == day].iloc[0]

# Insert Values

day\_word = row["Day.in.Words"]

gift = row["Gift.Item"]

item\_num = row["item\_amount"]

verb = row["Verb"] if pd.notna(row["Verb"]) else ""

adjective = row["Adjective"] if pd.notna(row["Adjective"]) else ""

location = row["Location"] if pd.notna(row["Location"]) else ""

# Choose plural or singular

if day > 1:

gift\_word = row["Gift.Plural"]

elif day == 1 and verb != "":

item\_num = "an"

gift\_word = gift

else:

gift\_word = gift

# Make Phrase

line = f"On the {day\_word} day of Christmas, my true love gave to me: {item\_num} {adjective} {gift\_word} {verb} {location}"

# Cleanup

line = " ".join(line.split())

line += "."

# Next Step

day = day -1

return line

def make\_phrase\_update(day, df=xmas):

"""

Sings the song based on the selected day.

Parameters

----------

DataFrame containing columns: 'Day' which is a number

Returns

-------

Text describing the gift for that day.

"""

# restrict dataframe to days <= selected day

sub = df[df["Day"] <= day].copy()

# Ensure correct order (day down to 1)

sub = sub.sort\_values("Day", ascending=False)

def make\_line(row):

"""Create a single line for one day."""

gift = row["Gift.Item"]

gift\_word = row["Gift.Plural"] if pd.notna(row["Gift.Plural"]) else gift + "s"

item\_num = row["item\_amount"]

verb = row["Verb"] if pd.notna(row["Verb"]) else ""

adjective = row["Adjective"] if pd.notna(row["Adjective"]) else ""

location = row["Location"] if pd.notna(row["Location"]) else ""

line = f"{item\_num} {adjective} {gift\_word} {verb} {location}"

return " ".join(line.split())

# use .map() on the rows

lines = sub.apply(make\_line, axis=1).tolist()

# fix the final "and" on the last line if >1 day

if len(lines) > 1:

lines[-1] = "and " + lines[-1]

# header

day\_word = df.loc[df["Day"] == day, "Day.in.Words"].iloc[0]

header = f"On the {day\_word} day of Christmas, my true love gave to me:"

# combine lines into a single verse

verse = header + "\n" + "\n".join(lines) + "."

return print(verse)

def make\_phrase\_update(dataset, day):

"""

Sings the song based on the selected day and dataframe. The song can be sung as either a single line or decending from the day selected.

Parameters

----------

DataFrame containing columns: 'Dataset', 'Day'

Returns

-------

Text describing the gift for that day.

"""

#Add the column

item\_amounts = [

'a', 'two', 'three', 'four', 'five', 'six',

'seven', 'eight', 'nine', 'ten', 'eleven', 'twelve'

]

dataset['item\_amount'] = item\_amounts

# Select the day

row = dataset.loc[dataset["Day"] == day].iloc[0]

# Insert Values

day\_word = row["Day.in.Words"]

gift = row["Gift.Item"]

item\_num = row["item\_amount"]

verb = row["Verb"] if pd.notna(row["Verb"]) else ""

adjective = row["Adjective"] if pd.notna(row["Adjective"]) else ""

location = row["Location"] if pd.notna(row["Location"]) else ""

# Choose plural or singular

if day > 1:

gift\_word = row["Gift.Plural"]

elif day == 1 and verb != "":

item\_num = "an"

gift\_word = gift

else:

gift\_word = gift

dataset["Gift.Plural"] = dataset["Gift.Item"].map(pluralize\_gift)

# Make Phrase

line = f"On the {day\_word} day of Christmas, my true love gave to me: {item\_num} {adjective} {gift\_word} {verb} {location}"

# Cleanup

line = " ".join(line.split())

line += "."

# Store in Full.Phrase

xmas.loc[row\_index, "Full.Phrase"] = line

return line